

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

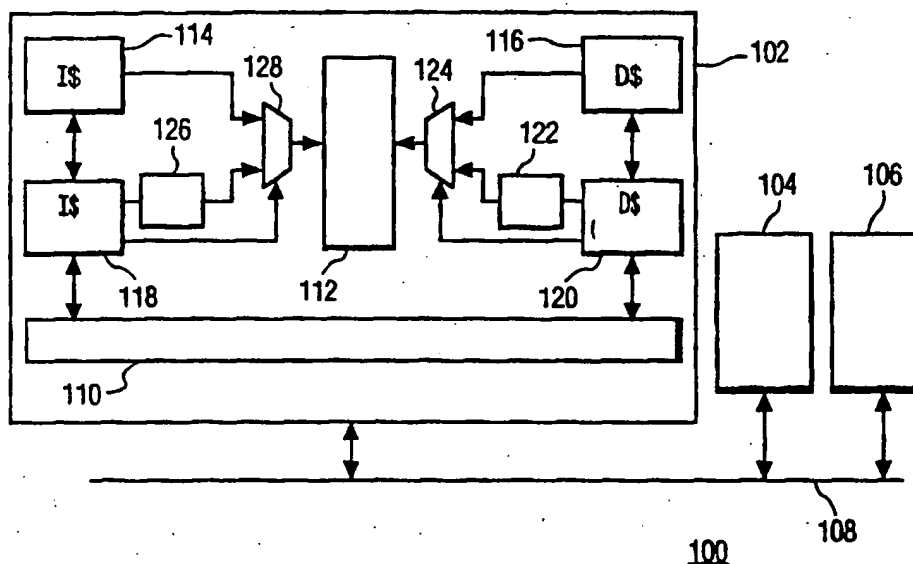
This Page Blank (uspto)



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : G06F 12/08, 9/30 // 13/00	A1	(11) International Publication Number: WO 99/32979 (43) International Publication Date: 1 July 1999 (01.07.99)
<p>(21) International Application Number: PCT/IB98/01549</p> <p>(22) International Filing Date: 5 October 1998 (05.10.98)</p> <p>(30) Priority Data: 08/995,820 22 December 1997 (22.12.97) US</p> <p>(71) Applicant: KONINKLIJKE PHILIPS ELECTRONICS N.V. [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL).</p> <p>(71) Applicant (for SE only): PHILIPS AB [SE/SE]; Kottbygatan 7, Kista, NL-164 85 Stockholm (SE).</p> <p>(72) Inventors: SIMOVICH, Slobodan; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL). ELTMAN, Brad; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).</p> <p>(74) Agent: KOPPEN, Jan; Internationaal Octrooibureau B.V., P.O. Box 220, NL-5600 AE Eindhoven (NL).</p>	<p>(81) Designated States: CN, JP, KR, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).</p> <p>Published <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i></p>	

(54) Title: EXTRA REGISTER MINIMIZES CPU IDLE CYCLES DURING CACHE REFILL



(57) Abstract

A CPU has an execution unit for operating on data under instruction control. A cache and a buffer register are coupled in parallel to an input of the execution unit. The buffer register supplies an information item, such as data or an instruction, to the execution unit upon the cache having completed a refill process.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LJ	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

Extra register minimizes CPU idle cycles during cache refill.

FIELD OF THE INVENTION

The invention relates to an electronic circuit comprising a CPU with a cache. The invention also relates to a method of supplying an information item, such as data or an instruction, to an execution unit of a CPU.

5

BACKGROUND ART

A CPU typically has one or more cache memories arranged between the data and instruction inputs of its execution unit on the one hand and the port for connection to main memory. The caches compensate for the difference in speed between the processing in the CPU and the fetching of data and instructions from main memory. The successful operation of the cache relies on the locality principle: program references to memory tend to be clustered in time and in logical space. Temporal clustering relates to the tendency to reference the same address more than once within a specific period of time. Spatial clustering relates to the tendency to fetch data or instructions from logically consecutive memory addresses. The data and instructions in the main memory are mapped into the cache in blocks of logically coherent addresses. Below, the term "information item" is used to refer to either data or an instruction within this context.

A cache read miss occurs when the CPU requests an information item that is not present in its cache. The cache has thereupon to retrieve the appropriate block from the main memory or the secondary cache and store it. During this cache refill, the execution unit is stalled. Various techniques are in use to minimize the number of clock cycles that the execution unit has to idle as a result of a cache refill.

For example, European patent application 0 543 487 A1 discusses the early-restart technique. As soon as the requested item arrives from main memory it is sent to the execution unit without waiting for completion of the writing of the entire block to the cache. A refinement of this early-restart is the out-of-order fetch. The out-of-order fetch lets the main memory skip all information items located at addresses logically preceding the requested item in the relevant block. The requested item is sent directly to the execution unit upon retrieval

while the remainder of the block is being retrieved looping around to fetch the items previously skipped.

European patent application 0 543 487 A1 also discusses an alternative technique that involves the following steps: If the CPU fetches data during a data cache fill and the requested data being fetched is part of the memory block being currently filled, the data is retrieved and returned to the execution unit simultaneously with its writing into the cache, if the data has not been written into the cache. If the data has been written into the cache, the data is retrieved and returned to the execution unit at the next read cycle.

Also see, e.g., "MIPS RISC Architecture", Gerry Kane and Joe Heinrich, Prentice Hall, 1992, notably Chapter 5, page 5-5. In the implementations of MIPS processor architectures, e.g., the R2000 and R3000, a typical sequence of events occurring after a cache miss are the following. On a cache miss, the processor reads one word from memory and stalls while the designated blocks in the cache are refilled. After the refill has been completed, missed information items are retrieved from the cache and are supplied to the processor's execution unit to resume processing. For general background information on the MIPS architecture, also see, e.g., "Structured Computer Organization", A.S. Tanenbaum, Prentice Hall International Editions, third edition, 1990, especially pp. 472-487.

OBJECT OF THE INVENTION

The advantages of early restart are limited if the execution unit processes the requested item faster than the cache can complete the refill. In the latter case, the execution has to idle after processing the item that was received directly until the cache has been refilled.

The alternative technique in the prior art reference discussed above addresses the problem of reducing the number of idle cycles of the execution unit while the cache is being refilled. This prior art reference does not address the problem of reducing the number of idle cycles when the refill has been, or nearly has been, completed. It is an object of the invention to increase performance of the processor by reducing the number of idle cycles substantially near completion of the cache refill.

SUMMARY OF THE INVENTION

To this end, the invention provides an electronic circuit comprising a CPU, an input for receipt of an information item, and a cache between the input and an execution unit of the CPU. The execution unit is operative to process the item. The circuit further comprises a buffer between the input and the execution unit, and a controller connected to the buffer. The

controller controls the storing of the information item in the buffer and the supply of the item to the execution unit substantially near completion of a cache refill.

The inventor proposes to use a temporary buffer in order to prevent the CPU from idling at least during the step wherein an item is being retrieved from the cache upon completion of the refill. The item is provided from the buffer instead of from the cache near completion of the refill. In this way, at least one clock cycle is saved per cache miss, since the buffer register is not address-controlled like a cache.

The circuit of the invention can use the buffer in combination with a main memory capable of early-restart and out-of-order fetch as mentioned above. The early restart/out-of-order fetch allows reducing the number of CPU idling cycles preceding the cache refill, and the buffer register in the invention reduces the number of CPU idling cycles after the cache has been refilled or has nearly completed refilling.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is explained below in further detail and by way of example with reference to the accompanying drawings, wherein:

Fig.1 is a block diagram of a circuit of the invention; and

Fig.2 is a diagram illustrating part of a cache controller's finite state machine.

PREFERRED EMBODIMENTS

Fig.1 is a functional block diagram with main components of an electronic circuit 100 according to the invention. Circuit 100 comprises a CPU 102, a bus controller 104 and a main memory 106 interconnected via a bus 108. CPU 102 has a bus interface 110, an execution unit 112, an instruction cache 114, a data cache 116, an instruction cache controller 118 and a data cache controller 120 for control of caches 114 and 116, respectively. CPU 102 further comprises a buffer register 122. In this example, buffer 122 and data cache 116 are arranged in parallel between controller 120 and unit 112. Buffer 122 and cache 116 are coupled to a data input of unit 112 via a multiplexer 124 that is controlled by controller 120.

Controller 120 controls buffer 122 to store data so that it can be supplied to execution unit 112 substantially at or near completion of a refill of cache 110 upon a read miss. In this manner, at least one clock cycle is saved during which unit 112 needs not idle. Buffer 122 supplies one or more data items to unit 112, e.g., when cache 116 is preparing for the first read cycle after the refill. The cache read cycle and the supply via buffer register 122 thus overlap in time.

Buffer 122 stores at least the data that was requested by CPU 102 and that caused the cache read miss. Buffer 122 may also store further data items at the addresses logically successive to the address of the requested data. Cache 116 also stores the requested data and the logically next data in order to comply with the cache's locality principle mentioned above.

The instruction path between cache 114 and unit 112 contains a similar buffer 126 and multiplexer 128 arrangement as the data path between cache 116 and unit 112 and it is controlled by instruction cache controller 118 in a similar way. Typically, the sequence of instructions during processing is known in advance. Conditional branching operations may occur in some software applications, but between two branching instructions the instruction stream is linear and known. If the buffer is a FIFO and stores two or more instructions, the supply of sequential instructions to unit 112 could start from buffer 126 before completion of the instruction cache refill and the start of the cache read cycle, or even substantially well before the completion of the instruction cache refill. To this end, controller 118 has to keep track of extent to which the address block has been mapped from main memory 106 to cache 114.

Fig.2 is a diagram of a part of a finite state machine 200 in cache controller 120. State machine 200 is explained in the following.

A stall cycle in a pipelines processor such as the MIPS 3000 is a cycle wherein the CPU waits for some event without doing useful work. For background information on the pipeline in the MIPS 3000, see, e.g., "MIPS RISC Architecture", Gerry Kane and Joe Heinrich, Prentice Hall, 1992, especially Chapter 1. The invention reduces the number of stall cycles upon a cache miss. In the MIPS 3000 the sequence of events upon a cache miss is the following. First, the designated blocks in the cache are refilled. Then, after the refill is completed, missed data is read from the cache and supplied to the execution unit. During both the refill phase and the cache read phase, the execution units is stalled. In the invention, however, execution unit 112 is not stalled during the cache read phase. Read data is stored temporarily in buffer 122 during the refill and is supplied to execution unit from buffer 122 and not from cache 116. In this manner at least one clock cycle is saved per individual cache miss. This is illustrated by state machine 200 that comprises the following transitions. Transition 202 from "read" to "refill" corresponds to a cache miss. Transition 204 from "read" to "non-cache" corresponds to a non-cacheable access to main memory (or secondary cache) 106. Transition 206 between "refill" and "wait" corresponds to the refill being completed and the execution unit not being ready yet. Transition 4 between "non-cache" and "wait"

corresponds to a non-cacheable word being fetched from main memory 106, and execution unit 112 not being ready yet to accept it. Transition 210 between "refill" and "read" corresponds to the refill being completed and execution unit 112 being ready. Transition 212 between "non-cache" and "read" corresponds to a non-cacheable word being fetched from main memory and execution unit 112 being ready. Transition 214 between "wait" and "read" corresponds to execution unit 112 being ready to accept the data. During the "refill" and "non-cache" states CPU 102 places the requested data in buffer 122 and communicates the data to execution unit 112 during transitions 210, 212 and 214.

CLAIMS:

1. An electronic circuit comprising a CPU having:
 - an input for receipt of an information item (110);
 - an execution unit (112) for processing the information item;
 - a cache (114 or 116) between the input and the execution unit;
 - 5 - a buffer (126 or 122) between the input and the execution unit for storing the item; and
 - a buffer controller connected to the buffer for control of storing of the item in the buffer and of supplying the item to the execution unit substantially near completion of a cache refill.
2. The circuit of claim 1, wherein the buffer controller comprises a cache
10 controller (118 or 120).
3. The circuit of claim 1, wherein the cache comprises a data cache (116).
4. The circuit of claim 1, wherein the cache comprises an instruction cache (114).
- 15 5. An electronic circuit comprising a CPU having:
 - a data input (110) for receipt of data;
 - an instruction input (110) for receipt of an instruction;
 - an execution unit (112) for processing the data under control of the instruction;
 - 20 - a data cache (116) between the data input and the execution unit;
 - a data buffer (122) between the data input and the execution unit for storing the data;
 - a data buffer controller (120) connected to the data buffer for control of storing of the data in the data buffer and of supplying the data to the execution unit substantially near completion of a data cache refill;
 - 25 - an instruction cache (114) between the instruction input and the execution unit;
 - an instruction buffer (126) between the instruction input and the execution unit for storing the instruction; and

- an instruction buffer controller (118) connected to the instruction buffer for control of storing of the instruction in the instruction buffer and of supplying the instruction to the execution unit substantially near completion of a instruction cache refill.

- 5 6. A method of information processing with an electronic circuit that has an execution unit for processing an information item and a cache (114 or 116), the method comprising storing the item in a buffer and supplying the item to the execution unit substantially near completion of a cache refill.

1/1

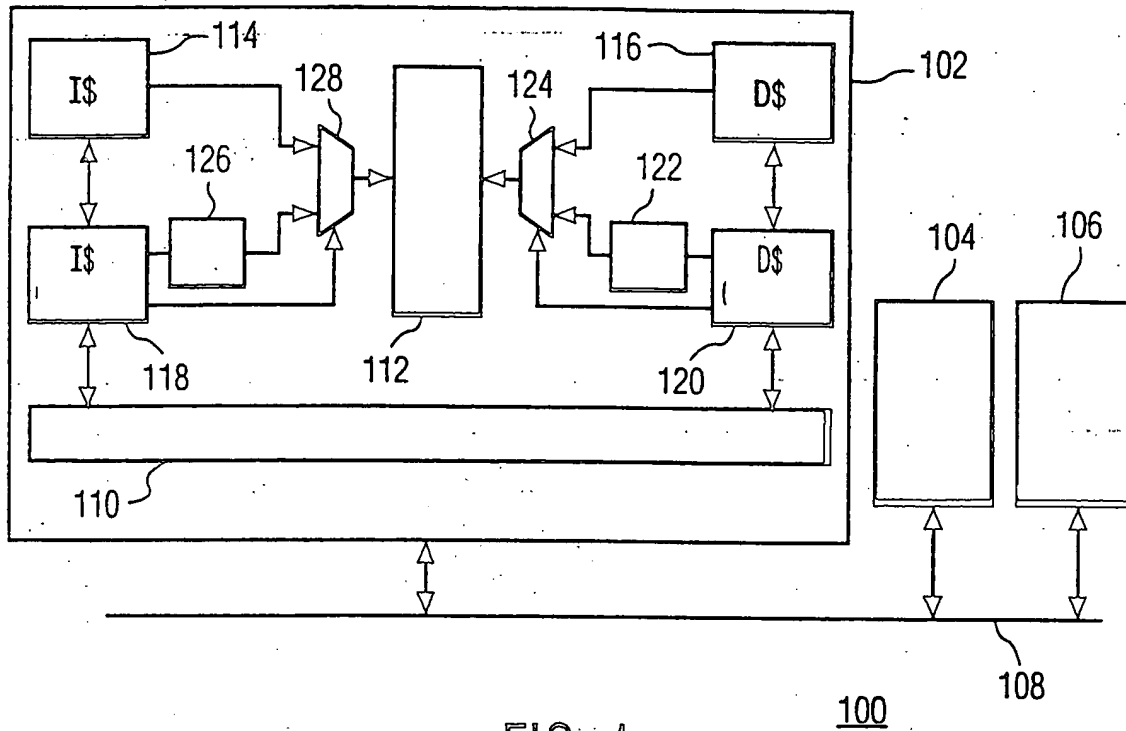


FIG. 1

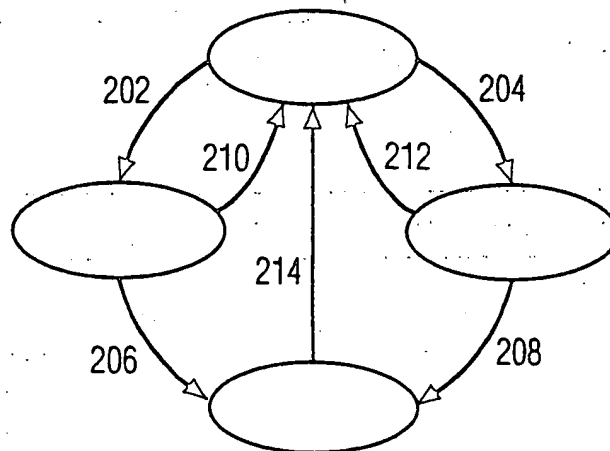


FIG. 2

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IB 98/01549

A. CLASSIFICATION OF SUBJECT MATTER

IPC6: G06F 12/08, G06F 9/30 // G06F 13/00
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC6: G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

DIALOG, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5353426 A (RAJIV N. PATEL ET AL), 4 October 1994 (04.10.94), column 3, line 3 - line 52, figures 2-3, claims 1-3,9-11, abstract	1-6
	--	
A	US 5388240 A (ULRICH OLDERDISSEN ET AL), 7 February 1995 (07.02.95), column 2, line 1 - line 35, figure 2, claims 1-3, abstract	1-6
	--	
A	US 5671444 A (HAITHAM AKKARY ET AL), 23 Sept 1997 (23.09.97), column 2, line 6 - line 60, figure 2, claims 1-3, abstract	1-6
	--	

☒ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "I" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- "&" document member of the same patent family

Date of the actual completion of the international search

6 May 1999

Date of mailing of the international search report

07 -05- 1999

Name and mailing address of the ISA/
Swedish Patent Office
Box 5055, S-102 42 STOCKHOLM

Authorized officer

Linus Wretblad

INTERNATIONAL SEARCH REPORT

International application No.

PCT/IB 98/01549

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5459852 A (HIROMASA NAKAGAWA ET AL), 17 October 1995 (17.10.95), column 4, line 1 - line 45, claim 1, abstract --	1-6
A	US 5179679 A (KENNETH D. SHOEMAKER), 12 January 1993 (12.01.93), column 2, line 1 - line 30, figure 4, claims 1-3, abstract -- -----	1-6

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/IB 98/01549

Patent document cited in search report			Publication date	Patent family member(s)	Publication date
US	5353426	A	04/10/94	JP 6318177 A	15/11/94
US	5388240	A	07/02/95	EP 0473804 A	11/03/92
				JP 6342400 A	13/12/94
				JP 8016883 B	21/02/96
US	5671444	A	23/09/97	US 5526510 A	11/06/96
US	5459852	A	17/10/95	JP 3214247 A	19/09/91
US	5179679	A	12/01/93	GB 2230120 A,B	10/10/90
				HK 112894 A	27/10/94
				JP 2293945 A	05/12/90
				SG 9094 A,G	10/06/94

This Page Blank (uspto)